UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/729,773	12/08/2003	Jeffrey B. Scott	I20 04984US	3422

128 7590 03/20/2008

HONEYWELL INTERNATIONAL INC.
101 COLUMBIA ROAD
P O BOX 2245
MORRISTOWN, NJ 07962-2245

EXAMINER	
WANG, JUE S	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
03/20/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/729,773	SCOTT, JEFFREY B.	
	Examiner	Art Unit	
	JUE S. WANG	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 November 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-12* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-12* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *08 December 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-12 have been examined.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 7-12 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

A. The following lacks antecedent basis in the claims:

i. Claim 7, "said validated import object" in line 6 because the claim only discloses "an import request to validate an import object" and "validating said import request".

B. The following claim language is not clear and indefinite:

i. As per claim 8, line 7, the phrase "locking said status of said existing object" is used. This limitation is not clearly understood because it is not clear if and how this step is performed when there is not an existing object when the determining step recited in lines 3-4 determines that the import object does not already exist as an existing object in a source control system? Furthermore, it is not clear if the outcome of the determining steps recited in lines 3-4 and 5 determines whether or not to lock the status of the existing object.

ii. As per claim 10, lines 5 and 6, the phrase "said existing object" is used. This limitation is not clearly understood because it is not clear if and how this step is performed when there is not an existing object when the determining step recited in lines 3-4 determines that the import object does not already exist as an existing object in a source control system?

iii. As per claim 11, line 3, the phrase "said existing object" is used. This limitation is not clearly understood because it is not clear if and how this step is performed when there is not an existing object when the determining step recited in lines 3-4 of parent claim 10 determines that the import object does not already exist as an existing object in a source control system?

Appropriate corrections are required.

Any claim not specifically addressed, above, is being rejected as incorporating the deficiencies of a claim upon which it depends.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claim 7 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cederqvist, "Version Management with CVS".

4. As per claim 7, Cederqvist teaches the invention as claimed, including a method comprising:

receiving an import request for an import object from an external source from a user (see page 69, section 13.2, paragraph 1);

validating said import request (see page 57, paragraph 3, page 69, section 13.2, paragraph 3 and page 95, section A.12, paragraph 4; EN: the import request is validated by checking if the file being imported already exists in the repository and has been locally modified which results in a conflict, and CVS refuses to check in a file if a conflict occurs until the conflict is resolved);

providing an import status (see pages 96-97, section A.12.2).

Cederqvist does not explicitly teach automatically checking-in said validated imported object. However, Cederqvist does teach that imported objects are checked in (see page 69, paragraphs 2, 5). Cederqvist also teaches that the usual procedure for adding files to a repository requires two separate commands: an "add" command to tell CVS that you want to version control the file and a "commit" command to actually check in the file (see page 43, section 7.1) It would have been obvious to one of ordinary skill in the art at the time of the invention that the imported object is automatically checked in via the "import" command because the import command checks in the imported object, but does not require a separate "commit" command to initiate the check in process for the imported object.

5. Claims 1-6 and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cederqvist, "Version Management with CVS", in view of Hammack et al. (US 6,449,624 B1, hereinafter Hammack).

6. As per claim 1, Cederqvist teaches a source control system, comprising:

an import function operable on said processor to import an object from an external source (see page 69, section 13.2, paragraph 1);

a validation function operable on said processor to determine whether said object is eligible for automatic check-in (see page 96, paragraph 1, EN: files that are not ignored are eligible for importing. While Cederqvist does not explicitly teach that imported files are automatically checked-in, Cederqvist does teach that imported files are checked in (see page 69, paragraph 2). Furthermore, since adding files to the repository typically requires two separate commands: an "add" command to tell CVS that you want to version control the file and a "commit" command to actually check in the file (see page 43, section 7.1). It would have been obvious to one of ordinary skill in the art at the time of the invention that the imported object is automatically checked in via the "import" command because the import command checks in the imported object, but does not require a separate "commit" command to initiate the check in process for the imported object.); and

a check-in function to be performed automatically upon import, including determining a version number for said object (see page 96, paragraph 3).

Cederqvist does not teach that the source control system is for a process control system with a processor, where import, validation, and check-in functions are operable on the processor.

Hammack teaches a process control system with a processor and version control functions for process configurations such as import and check-in (see Figs 1, 4, 6-8, abstract, column 2, lines 29-67, column 8, line 33 – column 13, line 57, column 16, lines 19-57).

It would have been obvious to one of ordinary skill in the art at the time of the invention to incorporate the source control system of Cederqvist with the import function into the process control system of Hammack because it is desirable to provide version control for process configurations in a process control system since multiple process operators modifying the process configuration stored in a configuration database will lead to version control problems when one operator is unaware of the work done by another operator (see column 1, line 51 – column 2, line 15 of Hammack), and in particular, the import function taught by Cederqvist allows the process control system to import process configurations from third parties (see page 69 of Cederqvist).

7. As per claim 2, Cederqvist does not teach that said object defines a control strategy.

Hammack teaches that the object tracked in the version control database defines control strategies (see Fig 3, column 6, line 19 – column 7, line 41).

8. As per claim 3, Cederqvist does not teach that the system comprising at least one controller capable of being loaded with said control strategy by said processor.

Hammack teaches at least one controller capable of being loaded with said control strategy by said processor (see Fig 1, item 12, column 3, lines 54-59, column 6, lines 19-24).

9. As per claim 4, Cederqvist does not teach the system further comprises at least one client in communication with said processor.

Hammack teaches the system further comprises at least one client in communication with said processor (see Fig 1, column 3, lines 48-67, column 6, lines 19-24).

10. As per claim 5, Cederqvist does not teach said control strategy is distributed from said processor to said at least one client.

Hammack teaches that the control strategy is distributed from said processor to said at least one client (see column 6, lines 19-24).

11. As per claim 6, Cederqvist teaches a database accessible by a processor to store said object (see page 69, the repository is the database that stores imported objects).

12. As per claim 12, Cederqvist does not teach that said imported object defines a control strategy.

Hammack teaches that the object tracked in the version control database defines control strategies (see Fig 3, column 6, line 19 – column 7, line 41).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Cederqvist such that the imported object defines a control strategy as taught by Hammack because it is desirable to provide version control for process configurations which define control strategies in a process control system since multiple process operators modifying the process configuration stored in a configuration database will lead to version control

problems when one operator is unaware of the work done by another operator (see column 1, line 51 – column 2, line 15 of Hammack).

13. Claims 8-10 are rejected under 35 U.S.C. 103(a) as being unpatentable over Cederqvist, "Version Management with CVS", in view of Tichy et al., "RCS – A system for Version Control" (hereinafter Tichy).

14. As per claim 8, Cederqvist teaches said validating said import request comprises:

determining if said import object already exists as an existing object in a source control system (see page 97, lines 1-2 and 4-5);

Cederqvist does not teach determining if said existing object has a status of checked-in, determining if said user has permission to check-in; and locking said status of said existing object.

Tichy teaches a system for version control including determining if a user has permission to check-in (see page 12, paragraph 3), determining if an object is checked in when checking out an object (see pages 11-12, section 2.4, paragraph 3) and locking the status of an object when someone intends to edit it (see page 11, section 2.4, paragraphs 2, 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Cederqvist to determine if said user has permission to check-in as taught by Tichy because it restricts the set of people who would have update permission (see page 12, paragraph 3); and it would been obvious to have modify Cederqvist to determine if said existing object has a status of checked-in and to lock the existing object as taught by Tichy because it

prevents another person from depositing competing changes to the same revision while the existing object is being updated by imported object (see page 11, section 2.4, paragraphs 2 and 3).

15. As per claim 9, Cederqvist does not teach unlocking said status of said existing object, after said imported object has been automatically checking-in.

Tichy teaches removing the lock after a check-in (see page 11, section 2.4, paragraph 3).

16. As per claim 10, Cederqvist teaches said automatically checking-in said import object comprises:

determining if said import object already exists as an existing object in a source control system (see page 97, lines 1-2 and 4-5);

determining a new version number for a new version of said existing object (see page 96, paragraph 3);

checking-in said imported object as said new version using said new version number (see page 69, paragraph 2, page 96, paragraph 3); and

storing a comment in said source control system indicating an automatic check-in for said new version (see page 69, section 13.2, paragraph 1 and page 96, section A.12.1, specifically, the "-m message" option).

Cederqvist does not teach determining if a status of said existing object is locked.

Tichy teaches a system for version control including determining if a status of an object is locked during a check-in operation (see page 11, section 2.4, paragraph 3).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Cederqvist to determine if the status of the imported object is locked during a check-in operation as taught by Tichy because it prevents multiple sources from checking in changes on the same file (see page 11, section 2.4, paragraphs 2 and 3 of Tichy).

17. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Cederqvist, "Version Management with CVS", in view of Tichy et al., "RCS – A system for Version Control" (hereinafter Tichy), as applied to claim 10 above, further in view of Shiman et al. (US 2002/0019827 A1, hereinafter Shiman).

18. As per claim 11, Cederqvist and Tichy do not teach that determining said new version number for said new version comprises: determining an existing version number of said existing object; determining an import version number from said import object; setting said new version number to a minor increment of said existing version number, if said import version number is equal to said existing version number; setting said new version number to a major increment of said existing version number, if said import version number is less than said existing version number; and setting said new version number to said import version number, if said import version number is greater than said existing version number.

Shiman teaches a method for managing documents in a centralized document repository (see abstract), where a new version number is determined for files uploaded to the repository, comprising: determining an existing identify tag including existing version number (see Fig 4, Fig 7, step 2712, [0074], and [0196]); determining an upload version number from the uploaded

file (see Fig 4, Fig 7, step 2701, [0074], and [0193]), setting the version number to a minor increment of said existing version number, if said uploaded version number is equal to said existing version number (see Fig 7, steps 2710, 2712, and [0196]). Shiman does not explicitly teach setting the new version number to a major increment of said existing version number, if said uploaded version is less than said existing version. However, Shiman teaches setting the new version number to be a major increment when the Branch tag of the uploaded file does not match an existing tag (see Fig 4, Fig 7, steps 2703, 2704, [0074], [0078], [0194]). Similarly, it would have been obvious to one of ordinary skill in the art at the time of the invention that a new branch with a major increment of the version number is created if the uploaded version number is less than the existing version number because the uploaded file must be from a new branch of development. Shiman also does not explicitly teach setting the new version number to the uploaded version number, if the uploaded version number is greater than the existing version number. However, Shiman teaches setting the new version to the next available minor version not used when the uploaded version is greater than the existing version (see Fig 7, steps 2710, 2711, [0196]). It would have been obvious to one of ordinary skill in the art that setting the new version number to the next available minor version achieves the same result as setting the new version number to the version number of the uploaded file because the next available minor version must be greater than all currently used minor versions since minor versions are incremented by one each time the owner submits a new document (see [0078]).

It would have been obvious to one of ordinary skill in the art at the time of the invention to have modified Cederqvist and Tichy to perform determining an existing version number of said existing object; determining an import version number from said import object; setting said

new version number to a minor increment of said existing version number, if said import version number is equal to said existing version number; setting said new version number to a major increment of said existing version number, if said import version number is less than said existing version number; and setting said new version number to said import version number, if said import version number is greater than said existing version number as taught by Shiman because it allows the import function of Cederqvist to import objects that already have version numbers associated and to use the version number to appropriately integrate the imported object into the source control system by constructing a new version number based on the version number of the imported object.

Response to Arguments

4. Rejection of claims 1-12 under 35 U.S.C. 112, second paragraph:

5. As per the rejections of claims 1, 7, 8, 10, 11 under 35 U.S.C. 112, second paragraph, Applicant's arguments have been fully considered. Examiner found the argument persuasive for the rejections at paragraphs 3Bi-3Biv, 3Bvii-3Bviii, and the rejections at paragraphs 3Bi-3Biv, 3Bvii-3Bviii are hereby withdrawn. As per the rejections at paragraphs 3Bv and 3Bvi for claims 8 and 10, Examiner respectfully disagrees with Applicant and maintains the rejection. Examiner believes the use of the phrase "said existing object" in claims 8 and 10 is indefinite because of the limitation of "determining if said imported object already exists as an existing object in a source control system" as recited in claims 8 and 10. This limitation includes the possibility that there does not already exist an existing object in the source control system and it is not clear how

the recited steps associated with the existing object such as "locking said status of said existing object" in claim 8 and "determining a new version number for a new version of said existing object" in claim 10 can be performed when there is not an existing object to perform these steps on. Claim 11 as amended to depend on claim 10 also exhibits the same indefiniteness with the recitation of "said existing object" and therefore is also rejected under 35 U.S.C. 112, second paragraph.

6. Rejection of claims 1 and 7 under 35 U.S.C.§103(a):

7. As per independent claims 1 and 7, Applicant's argued that Cederqvist lacks the step of "automatically checking-in said validated import object" as recited in claim 7 and "a check-in function opearable on said processor to be performed automatically upon import" as recited in claim 1. Applicant's arguments have been fully considered and Examiner respectfully disagrees. Examiner stated in the Office Action dated 8/24/2007 that Cederqvist does not **explicitly** teach the imported files are **automatically** checked in, however Cederqvist teaches that the imported files are checked in (see page 69, paragraph 2 and paragraph 5, sentence 1 that states "Use the import command to check in source for the first time"). Furthermore, Examiner stated evidence in Cederqvist that would have led one of ordinary skill in the art at the time of the invention to have concluded that the imported files are automatically checked in even though the disclosure of Cederqvist does not explicitly state the automatic nature of the check in for an import. Cederqvist teaches that adding files to the repository typically requires two separate commands: an "add" command to tell CVS that you want to version control the file and a "commit"

command to actually check in the file (see page 43, section 7.1). In this typical fashion of adding files to the repository, the checking in of the files is not performed automatically since the commit command must be issued to perform the check in. In contrast, the use of the import command checks in the files and this import command does not require a separate "commit" command to initiate the check in operation, therefore, it would have been obvious for one of ordinary skill in the art to have concluded that the check in operation is performed automatically when the import command is used to check in the source.

Conclusion

8. **THIS ACTION IS MADE FINAL**. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

19. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on 571-272-3759. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Lewis A. Bullock, Jr./
Supervisory Patent Examiner, Art Unit 2193

Jue Wang
Examiner
Art Unit 2193